

# Assessment 1

## Python Challenges

1 **Python 1** 3 PTS

Complete the function according to its docstring.

PYTHON3.6

```
1 def max_lists(list1, list2):
2     '''Compares the elements of list1, and list2 (assumed to be the same
3     length),and returns the maximum value between the two, for each element.
4
5     Parameters
6     -----
7     list1, list2: list, list
8         (Of the same length)
9
10    Returns
11    -----
12    list
13    '''
14    list3 = []
15    for a,b in zip(list1, list2):
16        if a>=b:
17            list3.append(a)
18        else:
19            list3.append(b)
20    return list3
```

RESET INPUT

RUN TESTS



No Time Limit

8/8 attempted

Save And Exit

▼ HIDE TEST RESULTS

```
•
-----
Ran 1 test in 0.000s

OK
```

## 2 Python 2 3 PTS

Complete the function according to its docstring.

PYTHON3.6

```
2     '''Given a matrix encoded as a 2 dimensional python list (i.e. a list of
3     uniform length lists), returns a list containing all the values along the
4     diagonal starting at the index 0, 0. (Assumes that the matrix is nonempty
5
6     Parameters
7     -----
8     mat: list of lists
9
10    Returns
11    -----
12    list
13
14    Example
15    -----
16    >>> mat = [[1, 2], [3, 4], [5, 6]]
17    >>> get_diagonal(mat)
18    [1, 4]
19    '''
20    import numpy as np
21    mat = np.array(mat)
22    return list(mat.diagonal())
```

RESET INPUT

RUN TESTS



CODING

### ▼ HIDE TEST RESULTS

```
•
-----
Ran 1 test in 0.161s

OK
```

3 **Python 3** 3 PTS

Complete the function according to its docstring.

PYTHON3.6

```
2     '''Returns a new dictionary containing all the keys from d1 and d2 with
3     their associated values. If a key is in both dictionaries, the new value
4     the sum of the two values from d1 and d2.
5
6     Parameters
7     -----
8     d1, d2: dictionary, dictionary
9
10    Returns
11    -----
12    dictionary
13    '''
14    list1 = list(d1.keys())
15    d3 = d1.copy()
16    for k2 in d2.keys():
17        if k2 in list1:
18            d3[k2] = d1[k2]+d2[k2]
19        else:
20            d3[k2] = d2[k2]
21
22    return d3
```

RESET INPUT

RUN TESTS



CODING

▼ HIDE TEST RESULTS

```
•
-----
Ran 1 test in 0.000s

OK
```

4 **Python/Linear Algebra** 3 PTS

Complete the function according to its docstring.

PYTHON3.6

```
22     '''
23     n = len(A)
24     result = []
25     # iterate over the rows of A
26     for i in range(n):
27         row = []
28         # iterate over the columns of B
29         for j in range(n):
30             total = 0
31             # iterate ith row of A with jth column of B dot product
32             for k in range(n):
33                 # k implements [ith row][jth column] element-wise dot product
34                 total += A[i][k] * B[k][j]
35             # column j of row i
36             row.append(total)
37         # all columns j of row i completed
38         result.append(row)
39     # all rows done
40     return result
41
42
```

RESET INPUT

RUN TESTS



CODING

▼ HIDE TEST RESULTS

```
.
-----
Ran 1 test in 0.000s

OK
```

## Pandas Challenges

For each Pandas challenge, you will be dealing with a `DataFrame` that contains median rental prices in the US by neighborhood. The `DataFrame` has these columns: `Neighborhood`, `City`, `State`, `med_2011`, `med_2014`

You can download a the data as [rent.csv](#) if you would like to work with it locally. If you do work with **rent.csv** data make note that the tests for these challenges only use a small subset of the dataset when executing your function. So *don't worry* if the results expected by the tests are different than what you found using the whole dataset.

5 **Pandas 1** 3 PTS

Complete the function according to its docstring.

PYTHON3.6

```
1 def pandas_add_increase_column(df):
2     '''Adds a column to the DataFrame called 'Increase' which contains the
3     amount that the median rent increased by from 2011 to 2014.
4
5     Parameters
6     -----
7     df: Pandas DataFrame
8
9     Returns
10    -----
11    None
12    '''
13
14    df['Increase'] = df['med_2014'] - df['med_2011']
15    return None
16
```

RESET INPUT

RUN TESTS



CODING

▼ HIDE TEST RESULTS

```
•
-----
Ran 1 test in 0.006s

OK
```

6 **Pandas 2** 3 PTS

Complete the function according to its docstring.

PYTHON3.6

```
1 def pandas_max_rent(df):
2     '''Returns a new pandas DataFrame that contains every city and the highest
3     median rent from that city for 2011 and 2014.
4
5     Note that city names are not unique so the state is tracked as well.
6     For example, Portland, ME and Portland, OR are recognized as different.
7
8     Parameters
9     -----
10    df: Pandas DataFrame
11
12    Returns
13    -----
14    Pandas DataFrame:
15        Containing the columns: City, State, med_2011, med_2014
16    '''
17
18    grouped = df.groupby(['City', 'State'])
19    return grouped[['City', 'State', 'med_2011', 'med_2014']].max()
```

RESET INPUT

RUN TESTS



CODING

▼ HIDE TEST RESULTS

```
.
-----
Ran 1 test in 0.020s

OK
```

## Conceptual Questions

7 **Git** 3 PTS

- a) What is Git, and why do software developers use it? (1 pt)
- b) What is the 'staging area', or 'index' in git? (1 pt)
- c) What does the command 'git commit' do? (1 pt)

a) Git is a version control system that allows you manage and control a project or a set of files over time. Developers use it to keep track of changes in their code, work simultaneously with others on large projects, and test changes in code on branches before merging them with their main branch, often called 'master'.

b) The staging area lets developers add the changes they've made to code before fully 'committing' the code, or adding it to tracked changes in git.

c) "git commit -m 'your commit message'" will add the files in your staging area or index to your tracked changes. It records changes to the repository.



CONCEPTUAL

8 **Python datatypes** 3 PTS

a) Please give an example of a mutable and immutable datatype. (2 pts)

b) What will be displayed when **b** is printed in the code below? (1 pt)

```
In[1] a = [1, 'a', [2, 3], 4.5]
```

```
In[2] b = a
```

```
In[3] a[0] = -0.1
```

```
In[4] print(b)
```

a) Tuple is immutable datatype.  
List is mutable datatype.

b) [-0.1, 'a', [2, 3], 4.5]



CONCEPTUAL



